5

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/646,309	08/22/2003	Gregory M. Wright	SUN-P9042-SPL	9198

57960 7590 06/01/2007
SUN MICROSYSTEMS INC.
C/O PARK, VAUGHAN & FLEMING LLP
2820 FIFTH STREET
DAVIS, CA 95618-7759

EXAMINER	
CHEN, QING	

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
06/01/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/646,309	WRIGHT ET AL.	
	Examiner	Art Unit	
	Qing Chen	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 August 2003.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-39 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-39 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 22 August 2003 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This is the initial Office action based on the application filed on August 22, 2003.

2. **Claims 1-39** are pending.

Specification

3. The abstract of the disclosure is objected to because it exceeds 150 words in length. Correction is required. See MPEP § 608.01(b).

Claim Objections

4. **Claims 19 and 36** are objected to because of the following informalities:

- **Claims 19 and 36** contain a typographical error: "an combining mechanism" should read -- a combining mechanism --.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. **Claims 6, 15, 24, 31, 35, and 39** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 6, 15, 24, 31, 35, and 39 contain the trademark or trade name JAVA, JVM, and JNI. When a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of the 35 U.S.C. 112, second paragraph. *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, the use of a trademark or trade name in a claim to identify or describe a material or product would not only render a claim indefinite, but would also constitute an improper use of the trademark or trade name.

Claim Rejections - 35 USC § 101

7. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

8. **Claims 10-27 and 32-39** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 10-18 and 32-35 recite computer-readable storage medium as a claimed element. However, it is noted that the specification describes such computer-readable storage medium as embracing computer instruction signals embodied in a transmission medium (with or without a carrier wave upon which the signals are modulated) *(see Page 5, Paragraph [0017])*.

Consequently, the computer-readable storage medium can be reasonably interpreted as carrying electrical signals.

Claims that recite nothing but the physical characteristics of a form of energy, such as a frequency, voltage, or the strength of a magnetic field, define energy or magnetism *per se*, and as such are non-statutory natural phenomena. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 112-14 (1853). Moreover, it does not appear that a claim reciting a signal encoded with functional descriptive material falls within any of the categories of patentable subject matter set forth in § 101.

Claims 19-27 and 36-39 are directed to apparatus. However, the recited components of the apparatus appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. The claims are directed to functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. **Claims 1-4, 6, 7, 10-13, 15, 16, 19-22, 24, and 25** are rejected under 35 U.S.C. 102(b) as being anticipated by **Kwong et al.** **(US 6,289,506).**

As per **Claim 1**, Kwong et al. disclose:

- selecting a call to a native code method to be optimized within the virtual machine *(see Column 8: 32-35, "... if the programmer decides to try to improve performance, then at step 730, he may select some of the Java program methods on the candidate list from step 720 for optimization.")*;
- decompiling at least part of the native code method into an intermediate representation *(see Column 8: 38-43, "... a user may decide to de-compile earlier native compiled code back to bytecode format. The de-compile process may be used for instance when a user determines that the native compiled code does not present the desired performance and the user wants to revert the native compiled code back to Java bytecode.")*;

- obtaining an intermediate representation associated with the application running on the virtual machine which interacts with the native code method *(see Column 8: 55-57, "A programmer writes Java source code 805 and compiles the source code into Java bytecodes 815 with a Java compiler 810.")*;

- combining the intermediate representation for the native code method with the intermediate representation associated with the application running on the virtual machine to form a combined intermediate representation *(see Column 8: 35-38, "... the selected Java program methods are optimized and compiled into native processor code by a native Java compiler."; Column 9: 7-9, "At the Java Platform, the bytecodes are loaded 825 into memory and then verified for security before they enter the Java VM 840.")*; and

- generating native code from the combined intermediate representation, wherein the native code generation process optimizes interactions between the application running on the virtual machine and the native code method *(see Column 8: 46-47, "... a programmer may repeat these steps to further refine and optimize the program."; Column 9: 9-11, "Once the bytecodes are in the Java VM 840, they are interpreted by a Java interpreter 842 or turned into native machine code by the JIT compiler 844.")*.

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Kwong et al. further disclose:

- wherein selecting the call to the native code method involves selecting the call based upon at least one of: the execution frequency of the call *(see Column 4: 11-19, "An analysis tool may track the Java program methods entered and exited in memory, establish a relationship* *between parent and child methods called, record every called program method, and time spend in each method. In another embodiment, an analysis tool may keep track of the Java methods being loaded into memory along with active software executing on the system. A tuning tool may determine the most active classes and methods in a Java application and list possible candidates for native compilation. ")*; and the overhead involved in performing the call to the native code method as compared against the amount of work performed by the native code method.

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and Kwong et al. further disclose:

- wherein optimizing interactions between the application running on the virtual machine and the native code method involves optimizing calls to the native code method by the application *(see Column 8: 32-35, "... if the programmer decides to try to improve performance, then at step 730, he may select some of the Java program methods on the candidate list from step 720 for optimization.").*

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Kwong et al. further disclose:

- wherein optimizing interactions between the application running on the virtual machine and the native code method involves optimizing callbacks by the native code method into the virtual machine *(see Column 7: 9-12, "In order to maintain the state of the Java VM 430 and make system calls, the compiled Java code 440 may make calls 450 into the Java VM 430.").*

As per **Claim 6**, the rejection of **Claim 4** is incorporated; and Kwong et al. further disclose:

- wherein the virtual machine is a Java™ Virtual Machine (JVM™) *(see Figure 2: 212)*; and

- wherein combining the intermediate representation for the native code method with the intermediate representation associated with the application running on the virtual machine involves integrating calls provided by the Java™ Native Interface (JNI™) into the native code method *(see Column 5: 41-44, "A Java Native Interface (JNI) may exist with the Java VM 212. The Java Native Interface is a standard programming interface for writing Java native methods and embedding the Java VM into native applications.")*.

As per **Claim 7**, the rejection of **Claim 1** is incorporated; and Kwong et al. further disclose:

- wherein obtaining the intermediate representation associated with the application running on the virtual machine involves recompiling a corresponding portion of the application *(see Column 8: 48-50, "The process of monitoring and compiling bytecode/de-compiling native code may be repeated until the desired performance is obtained.")*.

Claims 10-13, 15, and 16 are computer-readable storage medium claims corresponding to the method claims above (Claims 1-4, 6, and 7) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-4, 6, and 7.

Claims 19-22, 24, and 25 are apparatus claims corresponding to the method claims above (Claims 1-4, 6, and 7) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1-4, 6, and 7.

As per **Claim 28**, Kwong et al. disclose:

- deciding to optimize a callback by a native code method into the virtual machine *(see Column 7: 9-12, "In order to maintain the state of the Java VM 430 and make system calls, the compiled Java code 440 may make calls 450 into the Java VM 430.")*;

- decompiling at least part of the native code method into an intermediate representation *(see Column 8: 38-43, "... a user may decide to de-compile earlier native compiled code back to bytecode format. The de-compile process may be used for instance when a user determines that the native compiled code does not present the desired performance and the user wants to revert the native compiled code back to Java bytecode.")*;

- obtaining an intermediate representation associated with the application running on the virtual machine which interacts with the native code method *(see Column 8: 55-57, "A programmer writes Java source code 805 and compiles the source code into Java bytecodes 815 with a Java compiler 810.")*;

- combining the intermediate representation for the native code method with the intermediate representation associated with the application running on the virtual machine to form a combined intermediate representation *(see Column 8: 35-38, "... the selected Java program methods are optimized and compiled into native processor code by a native Java* *compiler."; Column 9: 7-9, "At the Java Platform, the bytecodes are loaded 825 into memory and then verified for security before they enter the Java VM 840.")*; and

- generating native code from the combined intermediate representation, wherein the native code generation process optimizes the callback by the native code method into the virtual machine *(see Column 8: 46-47, "... a programmer may repeat these steps to further refine and optimize the program."; Column 9: 9-11, "Once the bytecodes are in the Java VM 840, they are interpreted by a Java interpreter 842 or turned into native machine code by the JIT compiler 844.")*.

As per **Claim 29**, the rejection of **Claim 28** is incorporated; and Kwong et al. further disclose:

- wherein the native code generation process also optimizes calls to the native code method by the application *(see Column 8: 32-35, "... if the programmer decides to try to improve performance, then at step 730, he may select some of the Java program methods on the candidate list from step 720 for optimization.")*.

As per **Claim 31**, the rejection of **Claim 28** is incorporated; and Kwong et al. further disclose:

- wherein the virtual machine is a Java™ Virtual Machine (JVM™) *(see Figure 2: 212)*; and

- wherein combining the intermediate representation for the native code method with the intermediate representation associated with the application running on the virtual machine involves integrating calls provided by the Java™ Native Interface (JNI™) into the native code method *(see Column 5: 41-44, "A Java Native Interface (JNI) may exist with the Java VM 212. The Java Native Interface is a standard programming interface for writing Java native methods and embedding the Java VM into native applications.").*

Claims 32, 33, and 35 are computer-readable storage medium claims corresponding to the method claims above (Claims 28, 29, and 31) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 28, 29, and 31.

Claims 36, 37, and 39 are apparatus claims corresponding to the method claims above (Claims 28, 29, and 31) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 28, 29, and 31.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. **Claims 5, 14, 23, 30, 34, and 38** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Kwong et al. (US 6,289,506).**

As per **Claim 5**, the rejection of **Claim 4** is incorporated; however, Kwong et al. do not disclose:

- wherein optimizing callbacks by the native code method into the virtual machine involves optimizing callbacks that access heap objects within the virtual machine.

Official Notice is taken that it is old and well known within the computing art to allow callbacks to access heap objects within the virtual machine. Applicant has submitted in the specification that JNI™ provides an interface through which native code can manipulate heap objects within the JVM™ in a platform-independent way *(see Page 2, Paragraph [0004])*. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein optimizing callbacks by the native code method into the virtual machine involves optimizing callbacks that access heap objects within the virtual machine. The modification would be obvious because one of ordinary skill in the art would be motivated to allow the implementation of a Java™ object to remain transparent to the native code.

Claims 14 and 23 are rejected for the same reason set forth in the rejection of Claim 5.

As per **Claim 30**, the rejection of **Claim 28** is incorporated; however, Kwong et al. do not disclose:

- wherein optimizing the callback by the native code method into the virtual machine involves optimizing a callback that accesses a heap object within the virtual machine.

Official Notice is taken that it is old and well known within the computing art to allow callbacks to access heap objects within the virtual machine. Applicant has submitted in the specification that JNI™ provides an interface through which native code can manipulate heap objects within the JVM™ in a platform-independent way *(see Page 2, Paragraph [0004])*. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein optimizing the callback by the native code method into the virtual machine involves optimizing a callback that accesses a heap object within the virtual machine. The modification would be obvious because one of ordinary skill in the art would be motivated to allow the implementation of a Java™ object to remain transparent to the native code.

Claims 34 and 38 are rejected for the same reason set forth in the rejection of Claim 30.

13. **Claims 8, 17, and 26** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Kwong et al. (US 6,289,506)** in view of **Kilis (US 5,491,821)**.

As per **Claim 8**, the rejection of **Claim 1** is incorporated; however, Kwong et al. do not disclose:

- wherein obtaining the intermediate representation associated the application running on the virtual machine involves accessing a previously generated intermediate representation associated with the application running on the virtual machine.

Kilis discloses:

- wherein obtaining the intermediate representation associated the application running on the virtual machine involves accessing a previously generated intermediate representation associated with the application running on the virtual machine *(see Column 2: 2-4, "If the* *selected changed facet affects the object itself, then the previous intermediate representation of the object is modified.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Kilis into the teaching of Kwong et al. to include wherein obtaining the intermediate representation associated the application running on the virtual machine involves accessing a previously generated intermediate representation associated with the application running on the virtual machine. The modification would be obvious because one of ordinary skill in the art would be motivated to not reprocess existing information *(see Kilis – Column 1: 40-43).*

Claims 17 and 26 are rejected for the same reason set forth in the rejection of Claim 8.

14. **Claims 9, 18, and 27** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Kwong et al. (US 6,289,506)** in view of **Evans et al. (US 5,805,899)**.

As per **Claim 9**, the rejection of **Claim 1** is incorporated; and Kwong et al. further disclose:

- determining a signature of the call to the native code method *(see Column 4: 11-19, "An analysis tool may track the Java program methods entered and exited in memory, establish a relationship between parent and child methods called, record every called program method, and time spend in each method. In another embodiment, an analysis tool may keep track of the Java methods being loaded into memory along with active software executing on the system. A* *tuning tool may determine the most active classes and methods in a Java application and list possible candidates for native compilation.").*

However, Kwong et al. do not disclose:

- determining a mapping from arguments of the call to corresponding locations in a native application binary interface (ABI).

Evans et al. disclose:

- determining a mapping from arguments of the call to corresponding locations in a native application binary interface (ABI) *(see Column 7: 29-33, "Shared object 114 provides global symbols to which other objects, such as dynamic executable 120, can bind at runtime. These global symbols are specified in mapfile 130 and describe an Application Binary Interface (ABI) of shared object 114.").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Evans et al. into the teaching of Kwong et al. to include wherein determining a mapping from arguments of the call to corresponding locations in a native application binary interface (ABI). The modification would be obvious because one of ordinary skill in the art would be motivated to describe the low-level interface between an application program and the operating system, its libraries, or components of the application program.

Claims 18 and 27 are rejected for the same reason set forth in the rejection of Claim 9.

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / QC
May 26, 2007

[illegible]

WEI ZHEN
SUPERVISORY PATENT EXAMINER